P. Kühtreiber, V. Pak, and D. Reinhardt. “A method like this would be overkill”: Developers’ Perceived Issues with Privacy-preserving Computation Methods. Proceedings of the 22nd IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2023

“A method like this would be overkill”: Developers’ Perceived Issues with Privacy-preserving Computation Methods

Patrick Kühtreiber
University of Göttingen
Göttingen, Germany
kuehtreiber@cs.uni-goettingen.de

Viktoriya Pak
University of Göttingen
Göttingen, Germany
viktoriya.pak@stud.uni-goettingen.de

Delphine Reinhardt
University of Göttingen
Göttingen, Germany
reinhardt@cs.uni-goettingen.de

***Abstract*—According to the European General Data Protection Regulation and the principle of Privacy-by-design developers should embed methods of privacy protection into their projects at an early stage. However, studies show that developers are either lacking the tools or the training to apply the appropriate methods. Hence, to ensure the development of privacy-preserving software it is important to understand developers’ issues with methods of privacy-preserving computation. To this end, we have sent a questionnaire to 407 participants with diverse backgrounds to investigate their perceptions of privacy in general and of the methods k-anonymity, differential privacy, homomorphic encryption, and secure multi-party computation in particular. We compared the results to developers’ issues on Stack Overflow. We observe that raising the awareness about these methods increases developers’ willingness to use them in the future. We also validate previously known privacy-related issues developers face. Including privacy-preserving methods in programming education and, thereby, raising developers’ awareness could therefore enhance the privacy protection of software products.**

***Index Terms*—Privacy-by-design, k-anonymity, developers, differential privacy, homomorphic encryption, secure multi-party computation**

I. Introduction

Privacy laws and regulations, such as the European General Data Protection Regulation (GDPR), require developers to follow Privacy-by-Design (PbD) guidelines [13] that improve the privacy guarantees of IT systems [18]. Developers can apply Privacy-preserving Computation (PPC) methods, such as k-anonymity, Differential Privacy (DP), Homomorphic Encryption (HE), and Secure Multi-party Computation (SMPC), to reach this goal. Theoretical research on these methods is plentiful; however, their application in practice remains limited [1], [3]. Research on their usability and understanding often focuses on the end-user [14], [22], [51]. However, it is equally important to understand problems regarding comprehension and utility encountered by developers, whose first priority is usually not to ensure user privacy [39]. Most developers do not feel responsible for privacy, have not received any formal privacy-related training, and are often only willing to deal with PbD when it is explicitly demanded [21]. Privacy is therefore often seen as an obstacle to overcome [38]. Moreover, privacy can be seen as a preventive concept, the benefits of which are not immediately apparent [33]. To protect privacy, developers often adopt a methodology based on a variety of factors that include organizational mandate, compatibility with existing work practices, and their personal attitudes towards the methodology itself [31]. Moreover, motivation and knowledge are also key factors regarding developers’ usage of certain mechanisms [1]. This can also be applied in privacy engineering.

We therefore investigate whether raising awareness and knowledge about privacy-preserving methods for developers with no or very little previous experience raises their willingness to use these methods. Another factor influencing developers’ attitude towards privacy and security is the size of the company they work in [8]. The smaller the company, the less likely developers are to engage in privacy-preserving behavior. We hence evaluate whether different factors correlate with other aspects of privacy-engineering. These factors include company size, privacy attitudes, requiring proof of privacy education, or providing opportunities for privacy training. In summary, we investigate the following research questions:

- **RQ1**: What are developers’ previous experiences with anonymization, encryption, and data perturbation?
- **RQ2**: Which factors impact developers’ privacy attitudes and education?
- **RQ3**: What issues do developers perceive regarding the implementation of PPC methods?
- **RQ4**: Does raising the inexperienced developers’ awareness of certain PPC methods influence the willingness to use them in future?

To investigate our research questions we have designed an online questionnaire after having investigated the interest in PPC methods on Stack Overflow (SO). We have chosen three PPC methods (DP, HE, and SMPC) based on a study [3] in which experts were asked about to these techniques. We have included k-anonymity because anonymization is the primary technique used by developers when they are aiming at protecting user privacy [25], [36]. Our detailed contributions are as follows:

- We provide insights about developers’ experience with

anonymization, encryption, and data perturbation based on an online questionnaire with 407 participants.
- We evaluate developers' potential future usage of PPC methods.
- We observe that previous experience and raised awareness contribute positively in the developers' willingness to apply PPC methods in future software projects.
- We identify gaps in PPC knowledge as well as the factors *lack of perceived usability* and *lack of perceived need* as primary obstacles in adoption of PPC methods.
- We identify future research directions and provide guidelines on how to support developers in the development of privacy-preserving software as well as future research directions based on our findings.

The remainder of this paper is structured as follows: We discuss related work in Sec. II. We then lay out our methodology in Sec. III, before presenting our results in Sec. IV. We discuss our findings in Sec. V and conclusions in Sec. VI.

II. Related work

Developers consider privacy-related issues primarily if there are new policies or regulations [25]. Furthermore, many developers lack the necessary privacy-related skills [29] and often do not see privacy as their responsibility [17], [38], [39]. Also, privacy policies are often written on a high level to be as general as possible and therefore lack concrete instructions for developers to follow [2]. This creates problems, as developers often find it difficult to implement abstract privacy guidelines into concrete projects [34]. Moreover, developers are often unaware that these privacy guidelines even exist and, thus, do not seek any advice on how to comply [7].

The organizational privacy climate is an important factor and developers reject privacy architectures that contradict established frameworks [6], [35]. Similarly, *usability* and *established programming environments* are important parameters for a positive programmer experience [27]. Furthermore, developers trust more experienced colleagues when it comes to adopting certain methodologies [50].

However, following PbD guidelines is not merely an issue of engineering or usability, but a holistic problem that needs to be engaged from various disciplines and viewpoints [48]. Privacy patterns [15], i.e., the reuse of standardized and established privacy-preserving practices, are the most researched privacy design strategy [12]. However, the value of those patterns has not been thoroughly evaluated. Other research highlights the potential of design workbooks to improve developers' awareness of privacy-related issues and PbD [49]. Developers who work in larger companies tend to care more about implementing privacy and security measures. Also, security tools are used more often than privacy tools [8]. In line with earlier research on organizational challenges [20], [47], another study found obstacles, such as lack of a formal process within an organization, that complicates the development of secure software [5]. Moreover, a study measuring developers' privacy attitudes and perceptions revealed mismatches between these perceptions and their actual behavior as well as the importance of data monetization [26].

A recent study shows that developers go to SO to ask about privacy policies, access control, etc. The authors suggest that a more user-friendly workflow is necessary to guide developers to privacy-friendly software engineering [44].

Another study on developers' privacy perceptions on Reddit found that developers ask for advice about data protection regulations, e.g., the notion of consent under the GDPR [28]. These studies hence focus on a more general understanding of developers' privacy and security attitudes. We, however, investigate issues with privacy-preserving methods and how the concrete PPC methods are or could be used by developers and what are potential obstacles.

Findings of a study with nine industry experts on the PPC methods DP, HE, and SMPC suggest that these methods need to be made more understandable and, thus, more usable for developers [3]. The participants were, however, not developers. We bridge this gap by asking developers directly about their perception of privacy-preserving programming methods, as well as their awareness and expectations thereof.

To the best of our knowledge, so far no study has been conducted to evaluate awareness and usability of methods for privacy-preserving software development with developers.

III. Methodology

We have conducted a mixed design study (quantitative and qualitative data). To this end, we have first analyzed existing postings in SO to identify issues and prominence of the PPC methods k-anonymity, DP, HE, and SMPC. Since we focus on developers' problems, we did not include any SO-subsites in our analysis. Our SO analysis was conducted in December 2022. Due to the limited number of PPC posts, we searched for them directly on SO using the built-in search function, thus searching not only in title but also in the body of the posts. Our search queries were: *[ppc method] is:question* for questions and *[ppc method] is:answer* for answers; [ppc method] stands for the four PPC-methods investigated in this study. Note, that we used different spellings and versions of the techniques, such as "differentially private", "multi-party", etc.

In a next step, we have conducted an online questionnaire-based study targeted at developers. The questionnaire has been approved by our data protection officer and has been conducted according to ethical standards. The questionnaire can be found online (https://owncloud.gwdg.de/index.php/s/OLVlrNqlfdqlcUM). It is divided into four parts. The first part is dedicated to the participant's current projects, which privacy techniques are in use (if any), and which PPC methods are known. The second part deals with their experience in anonymization, data perturbation, and encryption. The third part focuses on developers with little or no PPC experience, applicability, and understandability. The final part addresses their privacy education, privacy training and attitudes, and demographics. Questions are based on previous studies regarding developers' security and privacy attitudes [5], [6], [26], [30], [39], privacy engineering methodologies [35], and

	Questions	Answers	Σ
k-anonymity	4	8	12
DP	35	17	52
HE	62	70	132
SMPC	7	4	11

TABLE I: Questions, answers, and total amount of posts of the respective PPC method on SO

debugging practices [30]. We have mitigated threats to internal validity by pilot-testing the questionnaire ($n = 3$). The pilot study resulted in refined explanations of the PPC methods and led to improved wording of our questions. Following other studies [5], [26], [40]–[43], [46], threats to external validity were addressed by choosing Prolific to recruit developers for our study. The participants were financially rewarded (£9/h) and could withdraw from the study at any time. Participants on Prolific are comfortable answering questions in English and show no signs of reporting a high number of invalid questions, as opposed to, e.g., MTurk. Furthermore, participants on Prolific are more diverse in terms of the male/female ratio compared to other sampling platforms [19]. To ensure a high data quality and a correct pre-screening, participants need to verify their identity and are required to answer a set of questions on their Prolific profile before they are able to take part in any surveys. Thereby, we were able to pre-select participants based on previous programming experience. Moreover, Prolific has fewer bots compared to MTurk [45].

IV. Results

A. Stack Overflow analysis

As seen in Tab I, manual search for k-anonymity, Differential Privacy (DP), Homomorphic Encryption (HE), and Secure Multi-party Computation (SMPC), resulted in very small corpora. HE was the largest corpus followed by DP and k-anonymity. SMPC is the least discussed.

When comparing the two most prominent Privacy-preserving Computation (PPC) methods DP and HE, we observe that even if the total number of HE posts exceeds DP posts, in recent years, interest in DP surpassed HE (see Fig. 1). One reason is the rising number of available DP libraries since 2019. Another one is that most younger questions are Machine Learning (ML)-related and ask about Tensor Flow Federated in particular. Based on this high-level analysis, we hence learn that PPC methods are not discussed to a large degree and can therefore assume that they are not widely used. We further leverage the resulting questions and responses published in SO for the interpretation of the results obtained with our questionnaire in Sec. IV-G.

> Developers do not discuss PPC methods a lot on SO. Interest in DP increased in recent years and overtook HE, mainly due to discussions about ML and Tensor Flow Federated.

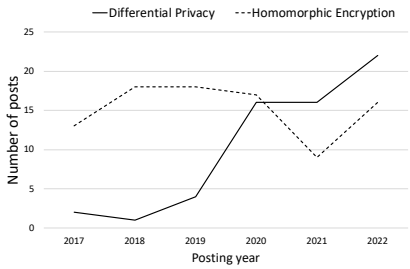

Fig. 1: Interest in DP surpassed HE in 2020.

B. Questionnaire

We have obtained quantitative results primarily by asking the participants' agreement to certain statements using a 5-point Likert scale (1="Strongly disagree", 5="Strongly agree"). We analyze the quantitative results using descriptive statistics. Due to the non-normality or our data, we explore correlations between the independent and dependent variables using Kruskal Wallis and Wilcoxon rank sum (for binary variables). We test correlations between dependent variables using Spearman's rho [16]. The qualitative results are analyzed using inductive coding by two researchers independently [11] (see Sec. IV-G) and statements are compared to the questions and answers obtained in our SO study detailed in Sec. IV-A.

1) Participants: We have received 433 complete responses and excluded 26 participants who contradicted themselves in their answers, by indicating a strong agreement to statements (1) anonymization is enough to protect privacy and (2) anonymization is *not* enough. That leaves 407 participants for our analysis. The average completion time was 12 minutes. Even though most participants were European (67%), the single largest contributing country in our sample was South Africa (21%). Participants who indicated that they taught themselves how to program predominantly also indicated that they work in ML (52% vs. 48% for the total sample) and Internet-of-Things (IoT) (56% vs. 55% total). Tab. II shows the diversity of our sample.

a) Processing of personal data: A majority of 81% said that they process at least some personal data and of those, 90% incorporate data-protection techniques. Of those that employ at least one privacy-preserving technique, most use encryption (85%), followed by anonymization (56%), and data perturbation (14%, multiple selections possible). Moreover, 80% of participants who indicated that they use anonymization, also indicated that they apply encryption. Likewise, 83% of participants using data perturbation also utilize encryption (see Sec. IV-C). As expected, there exists a correlation between company size and reported inclusion of data-protection techniques (Spearman's rho, $R = .152, p = .006$). For example, 96% working in a company with more than 1,000 employees include data-protection techniques, whereas it is only 78% of those working in smaller companies of nine employees or less. Furthermore, 72% indicated that they are at least partly responsible for incorporating data-protection measures. Participants indicated that dedicated security teams

Participant characteristics		
Gender	Male	72%
	Female	28%
Region of residence	Europe	73%
	Africa	23%
	Americas	2%
	Asia	1%
	Oceania	1%
Age	18-24	22%
	25-34	49%
	35-44	19%
	45-54	7%
	55+	3%
Programming education	Apprenticeship	2%
	School	7%
	Self-taught	22%
	Training/On the job	16%
	College/University	53%
Company size	<10	10%
	10-100	36%
	101-1000	29%
	>1000	23%
Programming experience in years	1-2	23%
	3-5	46%
	6-10	17%
	>10	14%

TABLE II: Participant characteristics show our diverse sample.

(29%), teammates (23%), or their supervisors (21%) are (also) responsible for data protection.

b) Privacy education: A minority (34%) indicated that they received privacy education, PbD training, etc., which confirms a previous study on the lack of security focus of developers recruited via Prolific [19], since security and privacy are intertwined topics. Among them, 54% reported that they visited privacy-related courses on their own initiative, 47% said that privacy engineering was part of their programming education, and 44% said that their employer offered privacy-related education (multiple selections possible). We observe no significant correlation between company size and privacy training offered by the employer. Programming education correlates significantly with privacy education. That means, participants who learned to program on the job or in college/university reported significantly more often to have received privacy education ($H_{(4)} = 18.38, p = .001$) with self-taught developers being the least likely. Participants' gender significantly correlates with having participated in at least one privacy training ($H_{(1)} = 5.84, p = .016$) and having visited it on their own initiative ($H_{(1)} = 5.58, p = .018$), with women reporting a higher agreement on both statements.

To gauge privacy attitudes and company requirements, we have asked our participants' agreement with the three privacy-attitude statements: (1) I need proof of privacy-related training for my job (**Proof**), (2) I am interested in privacy engineering (**Interest**), and (3) privacy is a priority in my projects (**Priority**). Developers working in IoT or ML reported a significantly higher agreement to all three statements (see Tab. III). Company size neither significantly correlates with our participants' privacy attitudes nor with the requirement of proof of privacy training. Most (90%) agree that there is a lack of privacy education in programming education. Consequently, 80% wish they had received more privacy training.

		Proof	Interest	Priority
Total	*M*	*3.3*	*3.8*	*3.7*
IoT	M	3.6	4.0	3.8
	$H_{(1)}$	12.00	8.30	8.31
	p	$< .001$	.004	.004
ML	M	3.6	4.0	3.9
	$H_{(1)}$	15.21	11.02	20.81
	p	$< .001$	$< .001$	$< .001$

TABLE III: Mean agreement (M) with the three privacy-attitude statements. Differences are significant for developers working in IoT or ML.

c) Troubleshooting: For privacy-related questions, 89% indicated that they search online (82% search engines, 52% SO, 32% Reddit), 53% consult colleagues, 30% ask their legal department, and 16% ask friends.

> Most participants process personal data and also safeguard them using predominantly encryption. Developers who work with IoT and ML report a higher interest in privacy. Almost all participants (90%) agree that there is a lack of privacy education in programming curricula.

C. Experience with privacy-preserving methods

Experience rating of the methods anonymization, encryption, and perturbation was done by indicating agreement with the statements: (1) it was easy to find solutions (**EasyFind**), (2) it was easy to choose the best fitting solution amongst them (**EasyChoose**), (3) the data protection technique negatively influences the data's usability (**NegImpact**), (4) performing anonymization/encryption/data perturbation is enough to ensure privacy (**Enough**), (5) it is easy to process data which has been anonymized/encrypted/perturbed (**EasyProc**), and (6) there are other measures necessary to protect users' privacy (**OtherNec**) (see Fig. 2). Data perturbation was the easiest to find (M=3.8, SD=1.0) and choose (M=3.7, SD=1.1) amongst the three methods. This was expected, as there exist more anonymization and encryption methods which makes choosing and finding appropriate solutions harder. Moreover, participants, who perform data perturbation, agree the least that this method is enough to protect an individual's privacy (M=3.2, SD=1.2) and consequently report the highest agreement regarding the question whether other measures are necessary (M=3.8, SD=1.2). Participants, who use encryption, report the highest negative impact on the data's utility (M=3.7, SD=1.2).

> Most developers who use data perturbation indicate that it is not enough to protect privacy; however it is the easiest method to find and choose. Encrypting data has the most negative effect on its usability.

D. Experience with PPC methods

We have asked our participants to rate their experience with each of the four PPC methods k-anonymity, DP, HE, and SMPC. Participants indicated their familiarity on a scale ranging from 1 ="I have never heard of it" to 4 ="I know it and have already used it". Against our expectations, 25% of our participants (across all regions) indicated to have used HE.

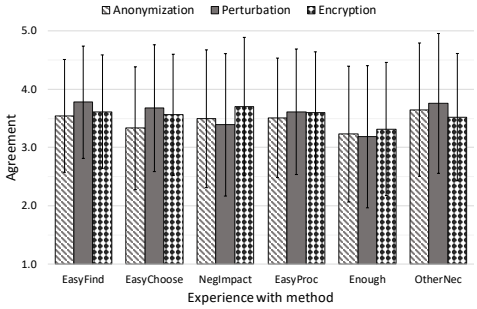

Fig. 2: Mean agreement and standard deviation for the six experience questions per data protection method

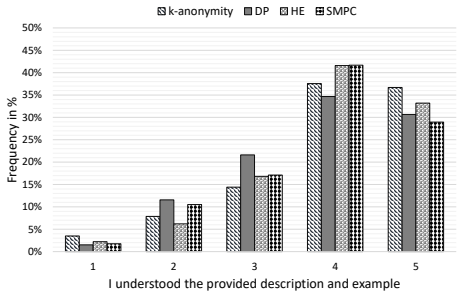

Fig. 3: Understandability rates are similar for all PPC methods (group **NEW** 1="Strongly disagree", 5="Strongly agree").

We hence must be careful when drawing conclusions from our findings regarding PPC experience. Our study hence especially focuses on identifying perceived obstacles for developers with very little or no experience with PPC methods, i.e., participants who indicated that they have only heard or never heard of a particular PPC method (28% k-anonymity, 23% DP, 31% HE, and 32% SMPC). We denote these participants as **NEW**.

E. Perceived understandability of PPC methods

NEW participants were shown an explanation of the PPC methods, which we have adapted and shortened (to avoid fatigue) from existing descriptions in scientific papers, textbooks, and publicly available lectures (k-anonymity [37], DP [51], HE [4], and SMPC [10]). Participants then rated the understandability of the description. The distributions are all left skewed, which indicates a good understandability (Fig. 3). There were no significant differences between developers who had visited a privacy training and those who had not.

F. Perceived suitability of PPC methods

Next, **NEW** participants were asked about their perception of the PPC method's usability: (1) It would make the data in the project I am currently working on unusable, (2) it seems easy to implement, (3) it contributes in improving the users' privacy, (4) I think that it could be applied in my current project, and (5) I think that it could be applicable in future projects. As expected, participants reported the highest concern about data's usability when DP is applied (M=3.3, SD=1.2).

Surprisingly though, our participants rated k-anonymity as being the hardest to implement (M=3.7, SD=1), even though it is arguably the easiest in practice. This can be due to our explanations, which might have simplified DP, HE, or SMPC. Generally, privacy improvements are expected of all methods. Also, comparing perceived applicability in current projects to future projects shows a significant increase in all PPC methods ($p < .001$). The mean agreement increases by 0.5 which suggests that the presented methods might be suitable for more developers than are currently using them.

> Developers who never used DP are suspicious about the data's utility after perturbation; however they believe that all PPC methods improve privacy and that they could be applied in future projects.

G. Reasons for rejecting the PPC methods

Participants, who indicated that they would not use a certain technique for current or future projects, were asked to provide a reason. Due to the novelty of our question, we have used inductive coding on those answers. Two researchers coded the answers independently from each other. Differences between coders revolved mainly around the number of topics. Once this was settled, inter-coder agreement was substantial [24] (Cohen's kappa 0.65). The remaining differences were discussed and resolved. The resulting themes are presented in the following. We further refine our results by providing examples from questions or answers based on our SO analysis. Statements from SO are highlighted with *italic* font.

- *Requirement of accurate data:* k-anonymity (67%), DP (52%), and SMPC (13%) fall into this category. In our questionnaire, statements include "We need the real data." [P312] and "[n]oise would be extremely detrimental for the project" [P36]. This can be confirmed by issues faced by developers on SO, stating that the data's utility "*suffers greatly for low values of epsilon*" [DPQ9], and is only given if you have "*copious amounts of data*" [DPA8]. Both statements are limitations of DP: a higher focus on privacy comes at the expense of the data's utility, hence, DP is only applicable if a lot of data is available.
- *Not secure enough:* k-anonymity (12%) and DP (5%) answers in our questionnaire indicate that "it may fail to protect against attribute disclosure" [P364] and that they are "[...] afraid of homogeneity attack" [P103]. The still existing problem of re-identification after k-anonymity is applied is also recognized by at least two SO users in the obtained corpus. [kAnonA2] states that "*that's not enough, because all k people identified by a distinct set of the different fields might be associated with the same value*", while [kAnonA4] indicates that "*you might still be able to identify a data subject indirectly through other information about them*".
- *Overkill:* k-anonymity (7%) and SMPC (11%) answers state that '[t]he effort required to do [k-anonymity] would probably not be worth it" [P401] or that "[a] method

like this one would be overkill" [P101]. Answers recommending SMPC in SO show various ways SMPC could be used, ranging from verifying that two secrets are the same [SMPCA1] to applications in conjunction with blockchain [SMPCA2]. Additional suggestions are secure ways to exchange [SMPCA4] or compare [SMPCA6] e-mail addresses or encrypting ML models [SMPCA7]. These examples highlight the ways in which SMPC could be used, which was not clear to our participants. The reason might be SMPC's perceived limited usability.

- *Utility:* This theme appeared in DP (28%). Developers are afraid that "[t]oo much noise could ruin a project, too little noise would make no sense to data protection" [P208] and regarding ML: "the data are altered and prevent the algorithm from learning correctly" [P72]. These concerns are valid, since DP leads to decreased data utility. Moreover, at least one developer on SO has the problem that presumably due to misconfiguration "*at some turning point, the privacy decreases and the accuracy decreases too*" [DPQ4]. However, DP can also increase accuracy in the general ML model, since "*sometimes the noise added [...] can help improve the accuracy a bit by helping prevent overfitting*" [DPA29].
- *Too complex:* DP (5%), HE (30%), and SMPC (24%) answers appear in this theme. Participants admit that they "[...] do not fully understand [DP]" [P163] and "[...] have yet to learn more about it in order to be able to properly apply [HE]" [P367]. HE's perceived complexity is also apparent when looking at the questions regarding HE on SO, as one author wrote: "*I have only basic knowledge in crypto, so I would like [to] use [HE] as black box as much as possible without putting to much effort in the mathematics behind it*" [HEQ11]
- *Efficiency:* Participants questioning HE's (30%) or SMPC's (13%) efficiency state "[HE] might potentially slow down the entire project" [P218] and "[SMPC] would take too long to implement [...]" [P362]. Comparing to SO, our participants correctly estimate the poor performance of HE in practice; one developer on SO wonders if its performance is "*the upper limit of the calculation speed of homomorphic encryption?*" [HEQ20].
- *Not compatible:* Finally, all PPC methods (k-anonymity 14%, DP 15%, HE 13%, and SMPC 11%) are perceived to not be compatible with current or future projects, saying that "[t]here are already rules that are applied in the project that contradicts the K-anonymity" [P106] and "[t]he data used are already anonymized [...]" [P18].

> DP and k-anonymity are primarily rejected because accurate data is needed; also, these methods are perceived as insecure. HE and SMPC are rejected due to perceived complexity and inefficiency.

H. Research questions

1) ***RQ1****: What are developers' previous experiences with anonymization, encryption, and data perturbation?* (1) Participants who perform anonymization have the hardest time to find and choose an appropriate solution, (2) data perturbation methods are the easiest to find and choose, (3) participants who encrypt data report the highest utility concerns, and (4) data perturbation is not widely used; however, those who do use it are generally satisfied with the results even though they indicate that additional measures are necessary. Issues regarding anonymization could be mitigated by increasing awareness about methods such as k-anonymity, as well as promoting usable libraries. Interest in privacy-preserving ML will presumably increase in the near future. This means that educators should incorporate DP within their programming curricula.

2) ***RQ2****: Which factors impact developers' privacy attitudes and education?* In our sample, company size neither significantly influences privacy-education requirements nor privacy training opportunities offered by the employer. Women visited privacy courses more often than men, so gender is a significant factor. Also, participants who learned to program in a structured environment report to a higher degree that they have received privacy education. Finally, working in IoT and ML correlates with interest in and requirement of privacy education. With the increase in privacy-awareness in the general public, software companies should offer privacy-related training on the job to their programmers. This way, trust in their software products can be increased.

3) ***RQ3****: What issues do developers perceive regarding the implementation of PPC methods?* Our participants primarily worry about the data's utility and the efficiency of the PPC methods. Most of the concerns are validated by comparing them with issues on SO. However, it is not always clear whether accurate data is really needed. As data minimization is one of the principles of the GDPR, developers and data scientists should continuously ask themselves if a certain data field is really necessary or whether it can be excluded in order to increase the data subjects' privacy.

4) ***RQ4****: Does raising the inexperienced developers' awareness of certain PPC methods influence the willingness to use them in future?* Increasing the awareness about PPC methods increases the willingness for future usage significantly, which is in line with previous research on non-PPC techniques [1].

V. Discussion

A. Developers and privacy engineering

Unsurprisingly, only a minority (34%) received any kind of privacy education. Consequently, most participants indicated that privacy education should be increased. These results are in line with previous research [29] and a possible source of non-privacy-preserving software practices. We observed no significant differences between IoT-developers, ML-developers, and the rest. Privacy education is more common when developers learned to program on the job or at college/university. Next to

search engines, developers refer to SO, which validates previous approaches of analyzing SO to understand developers' privacy issues [9]. Our participants generally do not agree that applying only one data protection technique is enough to protect privacy; however, some participants indicated that applying advanced privacy methods is "not necessary". Most participants apply data-protection methods and the rate increases with increasing company size, which confirms previous research [8]. One of the factors influencing methodology acceptance is understanding [32]. Teaching developers about these techniques can therefore positively influence future usage. However, organizational climate and external support are also factors in technology adoption, hence, organizations could enforce PPC methods' usage. Also, tools that support the implementation of PPC methods should be developed.

B. Experience with data protection methods

Most of our participants used encryption, which is unexpected, as previous research suggests that anonymization is more common [25], [36]. It was also indicated that perturbed data is easy to process and there is no negative impact on data utility, suggesting that developers who perform data perturbation are satisfied with the results and that raising awareness would positively impact the privacy guarantee of future projects. However, participants who perturb data also agree that further measures are necessary to protect personal data. Participants report the biggest issues with finding and choosing solutions to anonymize data. Raising awareness as well as providing accessible libraries to perform, e.g., k-anonymity would mitigate developers' problems in this regard.

C. Awareness of and experience with PPC methods

Reasons for not using k-anonymity and DP focus mainly on the limited data accuracy, which is problematic if you need exact data. Another reason is that both methods might not be secure enough. It is unclear whether developers actually need accurate data or whether they just prefer it. Either way, k-anonymity or DP cannot be applied in this case, as you always lose precision when applying data perturbation or anonymization. HE and SMPC are mainly rejected due to their complexity or because participants do not feel the need to use them. Performance and maturity issues are also mentioned. This underscores the point that PPC methods should be included in programming curricula.

D. Limitations

Choosing a panel provider ensures a high turnout rate and avoids biases introduced by the authors. Like all questionnaire-based studies, the results are however based on self-reported data. As a result, they should be completed with additional qualitative studies in future work. Validating self-reported experience with programming tasks could improve external validity. However, they would increase the time needed to complete questionnaire which could lead to fatigue and with recent advancements in AI, they can be solved by laypeople. Finally, while we did our best to ensure a concise and unbiased representation of the PPC methods' explanation, their respective framing might influence our participants' understanding and therefore the suitability rating.

VI. Conclusion

Understanding ways to make Privacy-preserving Computation (PPC) methods more usable for developers is key to privacy-preserving software engineering. The gap between privacy-engineering research and its application is wide and our study contributes in closing it. Our findings suggest that (1) teaching developers about PPC methods increases their willingness to use them in the future and (2) privacy is still an afterthought in programming education. Future work should focus on raising developers' awareness of and experience with PPC methods to ensure privacy-preserving software development. This can be done, e.g., by incorporating privacy training beyond encryption and anonymization into programming curricula. Moreover, companies that employ developers should provide opportunities for continuous privacy education. Furthermore, research should focus on building usable tools which aid developers in implementing privacy. These tools could be tailored, e.g., to the field of Internet-of-Things [23], but should focus on the implementation instead of a high level guidance. Developers benefit from PPC experience as, e.g., DP can be applied to Machine Learning (ML) applications. The correlation between working in ML and experience with DP and the trend on Stack Overflow show that ML developers are already incorporating DP into their projects. This suggests that DP is useful to ML developers. Currently, implementations of HE and SMPC lack in performance, but, technical advances suggest that these concepts will become valid which would also increase developers' experience with them.

References

[1] Yasemin Acar, Sascha Fahl, and Michelle L Mazurek. You are not your Developer, either: A Research Agenda for Usable Security and Privacy Research Beyond end Users. *IEEE Cybersecurity Development (SecDev)*, 2016.

[2] Yasemin Acar, Christian Stransky, Dominik Wermke, Charles Weir, Michelle L Mazurek, and Sascha Fahl. Developers Need Support, Too: A Survey of Security Advice for Software Developers. In *IEEE Cybersecurity Development (SecDev)*, 2017.

[3] Nitin Agrawal, Reuben Binns, Max Van Kleek, Kim Laine, and Nigel Shadbolt. Exploring Design and Governance Challenges in the Development of Privacy-preserving Computation. In *Proc. of the Conference on Human Factors in Computing Systems (CHI)*, 2021.

[4] Frederik Armknecht, Colin Boyd, Christopher Carr, Kristian Gjøsteen, Angela Jäschke, Christian A Reuter, and Martin Strand. A Guide to Fully Homomorphic Encryption. *Cryptology ePrint Archive*, 2015.

[5] Hala Assal and Sonia Chiasson. 'Think Secure from the Beginning' A Survey with Software Developers. In *Proc. of the Conference on Human Factors in Computing Systems (CHI)*, 2019.

[6] Oshrat Ayalon, Eran Toch, Irit Hadar, and Michael Birnhack. How Developers Make Design Decisions About Users' Privacy: The Place of Professional Communities and Organizational Climate. In *Companion of the 20th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*, 2017.

[7] Rebecca Balebako and Lorrie Cranor. Improving App Privacy: Nudging App Developers to Protect User Privacy. *IEEE Security & Privacy (S&P)*, 2014.

[8] Rebecca Balebako, Abigail Marsh, Jialiu Lin, Jason I Hong, and Lorrie Faith Cranor. The Privacy and Security Behaviors of Smartphone App Developers. *Proc. of the Workshop on Usable Security (USEC)*, 2014.

[9] Anton Barua, Stephen W Thomas, and Ahmed E Hassan. What are Developers Talking About? An Analysis of Topics and Trends in Stack Overflow. *Empirical Software Engineering*, 2014.

[10] Albrecht Beutelspacher, Jörg Schwenk, and Klaus-Dieter Wolfenstetter. *Moderne Verfahren der Kryptographie: Von RSA zu Zero-Knowledge*. Springer-Verlag, 2015.

[11] Virginia Braun and Victoria Clarke. *Thematic Analysis*. American Psychological Association, 2012.

[12] Julio C Caiza, Yod-Samuel Martín, Danny S Guamán, Jose M Del Alamo, and Juan C Yelmo. Reusable Elements for the Systematic Design of Privacy-friendly Information Systems: A Mapping Study. *IEEE Access*, 2019.

[13] Ann Cavoukian, Jules Polonetsky, and Christopher Wolf. Smartprivacy for the Smart Grid: Embedding Privacy into the Design of Electricity Conservation. *Identity in the Information Society (IDIS)*, 2010.

[14] Rachel Cummings, Gabriel Kaptchuk, and Elissa M Redmiles. "I need a better description": An Investigation Into User Expectations For Differential Privacy. In *Proc. of the 28th ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2021.

[15] Nick Doty and Mohit Gupta. Privacy Design Patterns and Anti-Patterns. *Proc. of the Trustbusters Workshop at the Symposium on Usable Privacy and Security (SOUPS)*, 2013.

[16] Andy Field and Graham Hole. *How to Design and Report Experiments*. Sage, 2002.

[17] Irit Hadar, Tomer Hasson, Oshrat Ayalon, Eran Toch, Michael Birnhack, Sofia Sherman, and Arod Balissa. Privacy by Designers: Software Developers' Privacy Mindset. *Empirical Software Engineering*, 2018.

[18] Dominik Huth and Florian Matthes. "Appropriate Technical and Organizational Measures": Identifying Privacy Engineering Approaches to Meet GDPR Requirements. 2019.

[19] Harjot Kaur, Sabrina Amft, Daniel Votipka, Yasemin Acar, and Sascha Fahl. Where to Recruit for Security Development Studies: Comparing Six Software Developer Samples. In *31st USENIX Security Symposium (USENIX Security 22)*, 2022.

[20] Sara Kraemer and Pascale Carayon. Human Errors and Violations in Computer and Information Security: The Viewpoint of Network administrators and security specialists. *Applied Ergonomics*, 2007.

[21] Patrick Kühtreiber, Viktoriya Pak, and Delphine Reinhardt. A Survey on Solutions to Support Developers in Privacy-preserving IoT Development. *Pervasive and Mobile Computing (PMC)*, 2022.

[22] Patrick Kühtreiber, Viktoriya Pak, and Delphine Reinhardt. Replication: The Effect of Differential Privacy Communication on German Users' Comprehension and Data Sharing Attitudes. In *Proc. of the 18th Symposium on Usable Privacy and Security (SOUPS)*, 2022.

[23] Patrick Kühtreiber and Delphine Reinhardt. Usable Differential Privacy for the Internet-of-Things. In *IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2021.

[24] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *Biometrics*, 1977.

[25] Tianshi Li, Elizabeth Louie, Laura Dabbish, and Jason I Hong. How Developers Talk About Personal Data and What It Means for User Privacy: A Case Study of a Developer Forum on Reddit. *Proc. of the ACM on Human-Computer Interaction (HCI)*, 2021.

[26] Dirk van der Linden, Irit Hadar, Matthew Edwards, and Awais Rashid. Data, Data, Everywhere: Quantifying Software Developers' Privacy Attitudes. In *International Workshop on Socio-Technical Aspects in Security and Trust (STAST)*, 2019.

[27] Jenny Morales, Cristian Rusu, Federico Botella, and Daniela Quiñones. Programmer eXperience: A Systematic Literature Review. *IEEE Access*, 2019.

[28] Jonathan Parsons, Michael Schrider, Oyebanjo Ogunlela, and Sepideh Ghanavati. Understanding Developers Privacy Concerns Through Reddit Thread Analysis. *arXiv preprint arXiv:2304.07650*, 2023.

[29] Mariana Peixoto, Dayse Ferreira, Mateus Cavalcanti, Carla Silva, Jéssyka Vilela, João Araújo, and Tony Gorschek. On Understanding How Developers Perceive and Interpret Privacy Requirements Research Preview. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, 2020.

[30] Michael Perscheid, Benjamin Siegmund, Marcel Taeumel, and Robert Hirschfeld. Studying the Advancement in Debugging Practice of Professional Software Developers. *Software Quality Journal*, 2017.

[31] Cynthia K. Riemenschneider, Bill C. Hardgrave, and Fred D. Davis. Explaining Software Developer Acceptance of Methodologies: A Comparison of Five Theoretical Models. *IEEE Transactions on Software Engineering*, 2002.

[32] Tom L Roberts, Michael L Gibson, and Kent T Fields. System Development Methodology Implementation: Perceived Aspects of Importance. *Information Resources Management Journal (IRMJ)*, 1999.

[33] Everett M Rogers, Arvind Singhal, and Margaret M Quinlan. Diffusion of Innovations. In *An Integrated Approach to Communication Theory and Research*. 2014.

[34] Awanthika Senarath and Nalin AG Arachchilage. Why Developers cannot Embed Privacy into Software Systems? An Empirical Investigation. In *Proc. of the 22nd International Conference on Evaluation and Assessment in Software Engineering (EASE)*, 2018.

[35] Awanthika Senarath, Marthie Grobler, and Nalin Asanka Gamagedara Arachchilage. Will They Use It or Not? Investigating Software Developers' Intention to Follow Privacy Engineering Methodologies. *ACM Transactions on Privacy and Security (TOPS)*, 2019.

[36] Swapneel Sheth, Gail Kaiser, and Walid Maalej. Us and Them: A Study of Privacy Requirements Across North America, Asia, and Europe. In *Proc. of the 36th International Conference on Software Engineering*, 2014.

[37] Vitaly Shmatikov. k-Anonymity and Other Cluster-Based Methods. *CS-380S Li, Li, Venkatasubramanian, ICDE*, 2007.

[38] Sarah Spiekermann and Lorrie Cranor. Privacy Engineering. *IEEE Transactions on Software Engineering*, 2009.

[39] Sarah Spiekermann, Jana Korunovska, and Marc Langheinrich. Inside the Organization: Why Privacy and Security Engineering is a Challenge for Engineers. *Proc. of the IEEE*, 2018.

[40] Mohammad Tahaei, Ruba Abu-Salma, and Awais Rashid. Stuck in the Permissions With You: Developer & End-User Perspectives on App Permissions & Their Privacy Ramifications. *arXiv preprint arXiv:2301.06534*, 2023.

[41] Mohammad Tahaei, Alisa Frik, Kami Vaniea, and Design Informatics. Deciding on Personalized Ads: Nudging Developers About User Privacy. In *Proc. of the 17th Symposium on Usable Privacy and Security (SOUPS)*, 2021.

[42] Mohammad Tahaei, Kopo M Ramokapane, Tianshi Li, Jason I Hong, and Awais Rashid. Charting App Developers' Journey Through Privacy Regulation Features in Ad Networks. *Proc. on Privacy Enhancing Technologies (PoPETS)*, 2022.

[43] Mohammad Tahaei, Kami Vaniea, Konstantin Beznosov, and Maria K Wolters. Security Notifications in Static Analysis Tools: Developers' Attitudes, Comprehension, and Ability to Act on Them. In *Proc. of the Conference on Human Factors in Computing Systems (CHI)*, 2021.

[44] Mohammad Tahaei, Kami Vaniea, and Naomi Saphra. Understanding Privacy-related Questions on Stack Overflow. In *Proc. of the Conference on Human Factors in Computing Systems*, 2020.

[45] Jenny Tang, Eleanor Birrell, and Ada Lerner. Replication: How Well Do My Results Generalize Now? The External Validity of Online Privacy and Security Surveys. In *Proc. of the 18th Symposium on Usable Privacy and Security (SOUPS)*, 2022.

[46] Daniel Votipka, Desiree Abrokwa, and Michelle L Mazurek. Building and Validating a Scale for Secure Software Development Self-efficacy. In *Proc. of the Conference on Human Factors in Computing Systems (CHI)*, 2020.

[47] Rodrigo Werlinger, Kirstie Hawkey, and Konstantin Beznosov. An Integrated View of Human, Organizational, and Technological Challenges of IT Security Management. *Information Management & Computer Security*, 2009.

[48] Richmond Y Wong and Deirdre K Mulligan. Bringing Design to the Privacy Table: Broadening "Design" in "Privacy by Design" Through the Lens of HCI. In *Proc. of the Conference on Human Factors in Computing Systems (CHI)*, 2019.

[49] Richmond Y Wong, Deirdre K Mulligan, Ellen Van Wyk, James Pierce, and John Chuang. Eliciting Values Reflections by Engaging Privacy Futures Using Design Workbooks. *Proc. of the ACM on Human-Computer Interaction (HCI)*, 2017.

[50] Shundan Xiao, Jim Witschey, and Emerson Murphy-Hill. Social Influences on Secure Development Tool Adoption: Why Security Tools Spread. In *Proc. of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*, 2014.

[51] Aiping Xiong, Tianhao Wang, Ninghui Li, and Somesh Jha. Towards Effective Differential Privacy Communication for Users' Data Sharing Decision and Comprehension. In *Proc. of the 41st IEEE Symposium on Security and Privacy (S&P)*, 2020.